TEXCAD User's manual

1 About TEXCAD

TEXCAD is a program for drawing or retouching {picture}s in LATEX. The features of environment {picture} are quite limited, but it presents the great advantage of requiring **no** add-on. You also enjoy the same fonts, macros, formulas as elsewhere in your LATEX document [1].
In addition, TEXCAD extends the original {picture} capabilities, even without any obligatory LATEX package, class or style sheet.
Furthermore, you can still switch on some supported add-ons, but are not obliged to.

1.1 Licence and distribution

1.2 New versions - TEXCAD on the Web

The latest version is (or should be, or might be) at the following URL: `http://texcad.sf.net`.

2 Presentation of TEXCAD — a few features

TEXCAD is rather thought for — but not limited to — simple figures drawn by hand. It lifts the annoyance of using a separate program just for drawing a little diagram with a box and a few arrows. The usual solution is to use a specific program, save the picture in EPS format, rescale, struggle to obtain the LATEX fonts, set the formulas in the picture, make it appear in the final format, which is never given. All this effort is a massive loss of time and nerves.

2.1 Files, I/O

The principle of TEXCAD is very simple: there is only one format, which is LATEX code. This means that you don't need to store your pictures in a format specific to TEXCAD and then "export" to a LATEX source in another file. In addition, it allows to rework a picture made with another program, e.g. GNUPLOT.

2.2 Integrating a TEXCAD picture into a LATEX file

All you have is to include your TEXCAD file (which is also just LATEX code) into your text, either via the `\input` command, or by inserting it directly. For instance, you save your picture under the name `toto.pic`, and in LATEX, put `\input toto.pic` where you want the picture to appear, or, alternatively, open `toto.pic` in a text editor, copy the full contents, paste it

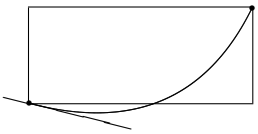
[1] This holds for previewing too: from TEXCAD version 4.2, you can insert your own definitions for previewing.

at that place.
As from TEXCAD 4.3, you can also select all items (Ctrl-A), copy them to the clipboard (Ctrl-C) and paste the picture's code into the LATEX code.
Here is an example of a TEXCAD picture insertion:

```
For obvious continuity reasons, the interpolation curve {\em will}
go outside the range if the slope has the inverse sign of the difference
of values (this excludes the cases with a zero slope or zero delta):
%
\begin{center}
%
%TeXCAD Picture [u3.tcp]. Options:
%\grade{\off}
%\emlines{\off}
%\epic{\off}
%\beziermacro{\on}
%\reduce{\on}
%\snapping{\on}
%\quality{8.00}
%\graddiff{0.01}
%\snapasp{1}
%\zoom{4.0000}
\unitlength 1.2pt
\linethickness{0.4pt}
\ifx\plotpoint\undefined\newsavebox{\plotpoint}\fi % GNUPLOT compatibility
\begin{picture}(81,41)(0,0)
\put(10,10){\framebox(70,30)[cc]{}}
\put(10,10){\circle*{2}}
\put(80,40){\circle*{2}}
\put(2,12){\line(4,-1){40}}
\qbezier(10,10)(59,-3)(80,40)
\end{picture}
%
\end{center}
%
```

will look like:

For obvious continuity reasons, the interpolation curve *will* go outside the range if the slope has the inverse sign of the difference of values (this excludes the cases with a zero slope or zero delta):



See the FAQ (4) about safely removing the TEX comments from the picture code.

2.3 More examples

You will find in the `Test_IO` directory, in the source archive, dozens of examples, mostly found on the Internet. Note that they have the `.tcp` extension instead of the default `.pic` extension. They are used for testing the TEXCAD Input/Output.

2.4 Compatibility

If there are commands unknown to TEXCAD, for instance color selection or whatever, they are **preserved and stored in the same place** of the picture file when TEXCAD saves the picture. Note that this program is compatible with the pictures saved by earlier versions, including the DOS "ancestor", TEXCAD 3.2.

3 How to use TEXCAD on MS Windows

The program behaves like usual Windows programs, so we assume you have a basic knowledge in that subject...

3.1 Installing and uninstalling

If you are a fan of watching ads during the hours an installation program may take, you'll be disappointed: TEXCAD doesn't need it. The program is wholly contained in TEXCAD.exe — not even a `DLL` around! So, if you rather dislike installations with lots of files scattered across your hard drive and Windows' system area, you'll be happy. You can even put TEXCAD.exe in a read-only drive, for instance a network drive without writing rights for users, since TEXCAD doesn't write any configuration file. Since the user settings are stored in the registry (HKEY_CURRENT_USER\Software\TeXCAD), the desinstallation requires running Uninstall_TeXCAD.bat logged as a super-user (in that case all TEXCAD information will be removed). As a "normal" user, you will just remove your own settings by this way.

3.2 Outlines

TEXCAD is a multi-document program, so you can work on several drawings — appearing each in a sub-window —, copy things from one to the clipboard, the paste them from the clipboard to another window.
The **mouse buttons** are defined so — this will be familiar to those who have used the DOS' TEXCAD — :

- The left button is for drawing or picking objects. This is the "Yes" or "+" button.
- The right button is for cancelling operations or unpicking objects This is the "No" or "-" button.

Another DOS' TEXCAD feature: you can move the mouse cursor with the arrow keys, e.g. for fine tuning.
We describe hereafter the functionalities of TEXCAD for Windows in the way the menus appear.

3.3 Files

TEXCAD is, like almost all common programs, file-based: there is a 1-to-1 relation between the picture being worked and a file name on a certain drive — with the exception of a new, unsaved picture which doesn't correspond to a file until it is saved the first time. Each time you save the picture, the file contains exactly the information currently displayed.
The commands "New", "Open", "Save", "Close", etc. should sound familiar to the user. On the menu's bottom there is a list of recently opened files.

3.3.1 Preview

The "Preview" command compiles the current picture with LaTeX and calls the installed DVI previewer - known to me are YAP, and DVIWin which has currently a much better rendering. If you have trouble with previewing, here is how it functions more in detail: TEXCAD writes, in the temporary directory [2], a `TeXCADpv.tex` file as well as a `TeXCADpv.bat` with the lines :

```
call latex TeXCADpv.tex
start TeXCADpv.dvi
```

[2] Alternatively, you can choose in the general options the **current** directory instead, but *caution*, the notion of "current" directory is floating if you have several pictures open stored in different places!

So, if you don't have the "latex" command available (in the PATH) you can create a batch file "latex.bat" with all workarounds you like...
If text boxes in your picture refers to specific definitions, you can set in the LaTeX tab in the picture options these definitions. The format is free, you can put there whatever you want, it will be inserted between the `\documentclass` line and the `\begin{document}` line.
NB: as from version 4.2 of TeXCAD, the preview uses by default the `\documentclass` syntax (LaTeX 2ϵ or later) in `TeXCADpv.tex` instead of the 2.09 `\documentstyle` syntax. You can still use the old syntax by selecting the appropriate general option.

3.4 Drawing

3.4.1 Objects defined by 1 point

It is a text in a broad sense. It can be also formulas between $ ... $, a sub-picture, an embedded EPS image, or whatever you put in your LaTeX document. Text is included in a `\makebox` with alignment options or in a raw `\put` command. You need one click for choosing the spot where the text appears.

3.4.2 Objects defined by 2 points

- **Rectangles**. You can choose between a frame or a filled rectangle. For non filled rectangles you can put a text in it — a dialog box will appear for entering it and set alignment.
- **Lines and vectors**. The function is obvious. In the picture options you can choose to have only the slope choice of the `\line` and `\vector` commands of LaTeX or any slope. TeXCAD finds out how to draw these line, independently of the choice of add-ons (none, emlines, pstricks,...).
- **Circles and discs**. These functions are quite obvious, aren't they ? Note that circles of a radius more than 20pt are not supported by pure LaTeX, but anyway TeXCAD, even in the mode without add-ons, is able to draw them with small line segments. You can choose their quality in the option panel. Same for discs: the LaTeX command is valid only up to 7.5pt; above that, TeXCAD fills the disc fractally with boxes.
- **Ovals** These are rather rounded rectangles. You can choose which side or corner will appear.

All figures determined by two points need drawing the mouse: press left button on first end and release it on the other end. Lines and vectors can also be chained: the further segments need clicking — in short: draw, click, click,... right-click to stop.

3.4.3 Objects defined by 3 points

For the moment, there are only Bézier splines in that category. Bézier curves are made by clicking the three points determining them, plus two per further curve — in short: click, click, click, click, click,... right-click to stop. Initially the splines are filled. You can change it with the "Change text or parameters" (3.6.1) menu command and clicking on the curve. In the general option panel, you can choose whether you want the filled curves first determined by

an amount of points fixed by TEXCAD or let LATEX find it, but this requires `\qbezier` — included for long in LATEX — instead of the old `\bezier`.

3.5 Lines (in a broad sense: also curves)

From TEXCAD 4.1 the line settings has been made "orthogonal" to the figures themselves, hence a new menu and the disappearance of some redundant figure choices. Some combinations are not meaningful (like arrows for boxes), or yet programmed in TEXCAD. At the end of this paragraph there is a synopsis of currently possible combinations.

3.5.1 Thin or thick

"thin" relates to the `\thinlines` macro (just the normal thickness) and "thick" to `\thicklines`, which displays lines (and curves) with a double thickness.

3.5.2 Patterns

You can choose and configure the following patterns: plain, dotted, dashed. Note that when you choose to recognize the `epic` environment, TEXCAD makes use of its `\dottedline` and `\dashline` macros.

3.5.3 Arrows

You have there the choice between "no arrow" (will output e.g. `\line` or `\bezier`), "head" (e.g. `\vector`), "both", arrow at both ends, "middle", an arrow on middle.

3.5.4 Synopsis of combinations figures / line settings

Line arrows \ pattern	plain, limited slope	plain, any slope	dotted	dash
no_ arrow	\line	\drawline \emline, %\emline	\dottedline	\dashline
head	\vector	%\vector	%\vector{dot}	%\vector{dash}
both	%\vector[b]{\line}	%\vector[b]	%\vector[b]{dot}	%\vector[b]{dash}
middle	%\vector[m]{\line}	%\vector[m]	%\vector[m]{dot}	%\vector[m]{dash}

Box pattern	plain	dotted	dash
	\framebox	%\dottedbox	\dashbox

Bezier arrows	
no_ arrow	\[[q]]bezier
head	%\bezvec
both	%\bezvec[b]
middle	%\bezvec[m]

The commands preceded by a '`%`' are *not* LATEX commands (or of the supported and switched on extensions like `epic`), but are to be understood by TEXCADonly. The code used by LATEX is in the lines between such a command (the command itself is seen by LATEX as a comment) and the line with `%\end`. See the FAQ (§4) Nr 3 for an example.

3.6 Editing

3.6.1 Change text or parameters

After having selected this menu entry, you can pick certain object for changing their contents and/or parameters:

Object	Content or parameter to change
Text	The text itself, alignment
Rectangle	Text, text alignment, dot length
Oval	Displayed edges or corners
Bézier curves	Number of dots, or automatic

3.6.2 Picking objects (individually, in an area, or all)

This how you start picking objects (*Pick objects*) in order to modify, copy or deleting them.

- If you click on an object with the left mouse button, you select ("pick") it, **individually**. If you click on an already selected object with the right button, you "unpick" it. The matching criterion is the distance of mouse cursor to the object (for a box: the distance to the frame or to the attached text).
- You can also select objects in an **area** by pushing the mouse button when the mouse cursor is in one corner of the area rectangle and releasing it on the opposite corner. Inbetween you see a dotted frame corresponding to the area's border. Same for unselecting an area: you do the press-move-release with the right button. The matching criterion is the presence of the *full* object in the area dotted frame.
- Finally you can *Select all* (the) *objects*, including the hidden ones TEXCAD didn't understand at loading, or *Unselect everything* of which was eventually selected previously.

3.6.3 Geometric operations

Once there are "picked" items, you can apply them a translation, a rotation, a symmetry (with a certain choice of axes) or an affine transformation. The translation needs a vector, you draw it as if it was a picture object. For the others a point (i.e. a mouse click) is needed.

3.6.4 Delete, cut, copy, save macro

All these operations need naturally that you have selected previously at least one object.

3.6.5 Paste, load macro

Paste and *load macro* operate on the same fashion: after choosing the menu item, you are asked to click on the spot where the lower left corner of the virtual rectangle containing the imported objects will land onto your picture.

3.7 View, toolbars

The "View" menu allows to choose which toolbar is shown. The "Drawing tools" toolbar corresponds to the "Drawing" menu, the "Line settings" toolbar to the "Line" menu.

3.8 Options

There are *General options* that contain various parameters you — as user — permanently prefer, and *Picture options* that are specific to the picture of the currently active subwindow. Note that you can in *General options* choose default parameters for every new picture. In the *Picture options* the compatibility ones need maybe some explanation. As the LaTeX world evolves some extensions to {picture} become obsolete, like emlines, other become almost standard. "Almost" is always the delicate point, since every LaTeX user or group of users has its own combination of add-ons, packages, fixes, patches, previewers, converters, devices and operating systems. For instance the PostScript-related add-ons are "almost" standard, but they are far of working in *every* situation, even if a PostScript printer is at the end of the line. Of course each package is a genial piece of work and I won't contest the quality of any. Simply, the philosophy behind TeXCAD is: if you like such or such package, use it, but we don't force you to use it. We always find a solution without it, at the price of a bigger picture file. After all the idea of TeX is to be DeVice Independent...

4 Frequently Asked Questions (FAQ)

1. *How can I put TeXCAD picture in my LaTeX file?*

   It's straightforward: just include or insert it - see §2.2.

2. *Do I have to add a special style file?*

   Only the ones you choose with the Picture Option panel (See "Compatibility (.sty)").

3. *TeXCAD puts plenty of useless comments in the file it writes. Why ?*

   Either they were already in the file as input, or they correspond to a command for TeXCAD which is not supported by a standard LaTeX macro or by a macro of an extension recognized by TeXCAD and enabled in the picture options.

   The "active" LaTeX code is between such a command (like: `%\dottedbox`) and a line consisting of: `%\end`. Please do **not** remove these comments! To get rid of them, you can activate a corresponding `.sty` option (picture options), or avoid using the corresponding figure (e.g. dotted lines if you don't want to use `epic`). Here is an example that shows how the principle works:

   With `epic` enabled you get:

   ```
   \dottedline(7,8.25)(45.25,19.5)
   ```

   Without `epic`, TeXCAD provides an emulation for LaTeX but considers itself only the code in the first commented line:

   ```
   %\dottedline(7,8.25)(45.25,19.5)
   \multiput(6.93,8.18)(.93293,.27439){42}{{\rule{.4pt}{.4pt}}}
   %\end
   ```

   Here is a more subtle case which shows that TeXCAD is able to provide *several levels* of emulation according to which style extensions are enabled:

(a) Nothing, pure LaTeX: a Bézier curve is drawn with sequence of small segments. Since the slopes of pure LaTeX are limited, some oblique segments are drawn via an emulation with a sequence of small horizontal segments:

```
%\qbezier(2,1.5)(4.563,2.438)(5.875,3.125)
\multiput(2,1.5)(.12527,.046929){8}{\line(1,0){.12527}}
\multiput(3.002,1.875)(.108492,.042532){6}{\line(1,0){.108492}}
\multiput(3.653,2.131)(.099481,.04073){6}{\line(1,0){.099481}}
\multiput(4.25,2.375)(.10965,.04693){4}{\line(1,0){.10965}}
\multiput(4.689,2.563)(.10208,.04542){3}{\line(1,0){.10208}}
\multiput(4.995,2.699)(.09559,.04412){3}{\line(1,0){.09559}}
\put(5.282,2.831){\line(1,0){.1804}}
\put(5.462,2.917){\line(1,0){.1717}}
\put(5.634,3.002){\line(1,0){.1631}}
\put(5.797,3.084){\line(1,0){.0783}}
%\end
```

(b) `epic` enabled but not `bezier`: the sequence of segments emulating the curve uses the `\drawline` command in `epic`:

```
%\qbezier(2,1.5)(4.563,2.438)(5.875,3.125)
\drawline(2,1.5)(3.002,1.875)(3.653,2.131)(4.25,2.375)(4.689,2.563)(4.995,2.699)
(5.282,2.831)(5.462,2.917)(5.634,3.002)(5.797,3.084)(5.875,3.125)
%\end
```

(c) `bezier` enabled, other options meaningless:

```
\qbezier(2,1.5)(4.563,2.438)(5.875,3.125)
```

As a conclusion for this question, please don't remove the TeX comments inside the TeXCAD picture text, unless you intend not to rework your picture again anymore. Even then, you will be able to do it but won't see everything on screen, or lose some structures or groupings. Note that you can remove harmlessly the comments before "`\begin{picture}`" since they only contain the switches for the TeXCAD picture options.

5 The authors

Georg Horn, Jörn Winkelmann	:	The original TeXCAD for DOS (1989-1994), in Pascal, up to v. 3.2
Gautier de Montmollin	:	Translation to Ada via P2Ada, TeXCAD 4.x system, Windows "hull"

I (Gautier) am the maintainer of the current project.
Contact me at `http://sf.net/users/gdemont/`.
Current TeXCAD Web page is `http://texcad.sf.net`.
The development project is also hosted at SourceForge: `http://sf.net/projects/texcad/`.

6 Acknowledgements — for tools, help, testing, ideas, remarks, asking questions

Note that all wishes are not yet concretized. Be patient, it is a only question of time! So, thanks to:

- David Botton, for his GNAVI R.A.D. system, in particular his wonderful GWindows library (URL: `http://sourceforge.net/projects/gnavi/`) and his help to this project.

- – Prof. Robert A.G. Seely (McGill), for his enthusiastic testing.
  – Bernd S. W. Schroeder (Louisiana Tech University), for suggestions and his review in MERLOT.
- Also thanks to ...
  – *User side:* Th. Bauer, Prof. Dr.-Ing. Helmut Bollenbacher, Prof. Carlos A. Cinquetti, Prof. Jean-Pierre Corriou, Maxwell Dondo, Zeno Enders, Roman Heinisch, Shul-John Li, Alessandro Rosa, Paolo Tommasini, Rune Tønnesen, Song Yu, Dr Yahya H. Zweiri.
  – *Programming side:* André van Splunter

7 How to build the program yourself

To build the sources, you need an Ada compiler, for instance the GNU Ada compiler (GNAT) that can be found there: `http://libre.adacore.com/`
Alternatively you can use the MinGW version: `http://mingw.org/`. In case there is an implementation for Linux, GNAT is available with the standard packages.
The project uses extensively Ada's modularity. The platform-dependent parts are confined to their minimum and the core of TEXCAD is totally portable across platforms and compilers, without conditional compilation – which doesn't exist in Ada by the way! The core includes:

- the memory storage of pictures
- transformations
- input / output
- object picking rules
- ... and even the display, which is possible in a platform-independent way through the magic of Ada's *generic programming*

As a test I build and run regularily the core with GNAT, on both Windows and Linux, as well as with another development system, ObjectAda 7.2.2 SE (free version), available at `http://www.aonix.com/`.

The Windows "skin" uses the open-source framework GWindows for the user interface.
All you need to do for building TEXCAD for Windows is to install a binary distribution of the **GNAT Ada compiler** for Windows platforms (from AdaCore or MinGW, links above). Then, go to the `TC_GWin` directory and run `"build_debug.cmd"`, or open `texcad_gwin.gpr` with GPS (the GNAT Programming Studio) and press F4.
NB: a snapshot of the **GWindows** library, ANSI mode, is included with TEXCAD's sources (in `TC_GWin/windows_stuff/`), so you don't need to install it! If for whatever reason you need to update GWindows, you'll find it there: `http://sourceforge.net/projects/gnavi/`

The principal directories are:

(root)	:	the core of TEXCAD as described earlier. There is a test program, Test_TC_IO, mainly an entropy test, that does a massive number of Load/Format-change/Save operations and checks if something gets lost, also in accuracy.
TC_GWin	:	the parts for the Windows version.
Test_IO	:	a few pictures, some of mine, some grabbed from the web, some output of GNUPlot. Test_TC_IO reads them and write copies, for checking consistency.

The full tree looks like: